

AD A053016

6

ISI/SR-78-10
March 1978

ARPA ORDER NO. 2223

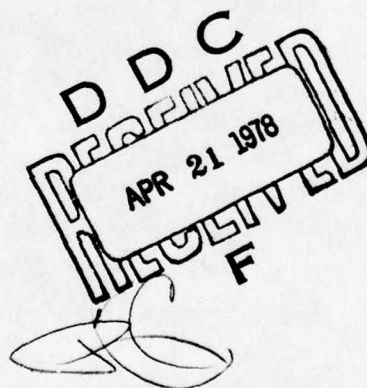


PROTECTION ERRORS IN OPERATING SYSTEMS:

A Selected Annotated Bibliography and Index to Terminology

AD No. _____
JDC FILE COPY

Jim Carlstedt



This document has been approved
for public release and sale; its
distribution is unlimited.

INFORMATION SCIENCES INSTITUTE

UNIVERSITY OF SOUTHERN CALIFORNIA



4676 Admiralty Way/Marina del Rey/California 90291
(213) 822-1511

ISI/SR-78-10

March 1978

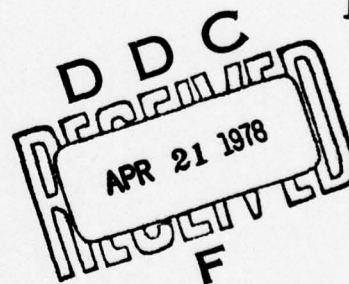
ARPA ORDER NO. 2223



PROTECTION ERRORS IN OPERATING SYSTEMS:

A Selected Annotated Bibliography and Index to Terminology

Jim Carlstedt



INFORMATION SCIENCES INSTITUTE

UNIVERSITY OF SOUTHERN CALIFORNIA



4676 Admiralty Way/Marina del Rey/California 90291
(213) 822-1511

THIS RESEARCH IS SUPPORTED BY THE ADVANCED RESEARCH PROJECTS AGENCY UNDER CONTRACT NO. DAHC15 72 C 0308, ARPA ORDER NO. 2223.

VIEWS AND CONCLUSIONS CONTAINED IN THIS STUDY ARE THE AUTHOR'S AND SHOULD NOT BE INTERPRETED AS REPRESENTING THE OFFICIAL OPINION OR POLICY OF THE UNIVERSITY OF SOUTHERN CALIFORNIA OR ANY OTHER PERSON OR AGENCY CONNECTED WITH IT.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

14 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ISI/SR-78-10✓	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 Protection Errors in Operating Systems: A Selected Annotated Bibliography and Index to Terminology•		5. TYPE OF REPORT & PERIOD COVERED 9 Research Report, 10
7. AUTHOR(s) 10 Jim/Carlstedt		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS USC/Information Sciences Institute✓ 4676 Admiralty Way Marina del Rey, CA 90291		8. CONTRACT OR GRANT NUMBER(s) DAHC 15 ⁷² C0308✓
11. CONTROLLING OFFICE NAME AND ADDRESS Advanced Research Projects Agency 1400 Wilson Blvd. Arlington, VA 22209		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS ARPA Order # 2223
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) -----		12. REPORT DATE February 1978 Mar 78
		13. NUMBER OF PAGES 12 48 p.
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release and sale; distribution unlimited. 15 DAHC 15-72-C-0308, ARPA Order-2223		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) -----		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) operating systems protection, security, bibliography		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) (OVER)		

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. ABSTRACT

➤ This report represents the current state of a bibliography on the subject of protection in computer operating systems. "Current state" means that the bibliography is incomplete; it is a byproduct of a research project in the field of protection, recently completed. The bibliography is being published in the belief that it may be useful as is, and that it might serve as the basis of a continuing effort to collect, annotate and index the more significant documents (reports, papers, articles, books, etc.) in the field. Ideally (especially in these days of computerized information bases and communication networks) workers in a research field will collaborate in developing and sharing their bibliographies--not only with simple annotations like this one but with more extensive comments and reviews. Perhaps this document can be a contribution in that direction and will stimulate owners of other "working" bibliographies to publish theirs. As noted below, this bibliography is online and may be accessed via the ARPANET.

(Defence Advanced Research Projects Agency network of interconnected scientific computers).

ACCESSION for	Write Section <input checked="" type="checkbox"/>
NTIS	B. R. Section <input type="checkbox"/>
DDC	
UNANNOUNCED	
JUSTIFICATION	
BY	DISTRIBUTION/AVAILABILITY CODES
	SPECIAL
A	

UNCLASSIFIED

INTRODUCTION

This report represents the current state of a bibliography on the subject of protection in computer operating systems. "Current state" means that the bibliography is incomplete; it is a byproduct of a research project in the field of protection, recently completed. The bibliography is being published in the belief that it may be useful as is, and that it might serve as the basis of a continuing effort to collect, annotate and index the more significant documents (reports, papers, articles, books, etc.) in the field. Ideally (especially in these days of computerized information bases and communication networks) workers in a research field will collaborate in developing and sharing their bibliographies--not only with simple annotations like this one but with more extensive comments and reviews. Perhaps this document can be a contribution in that direction and will stimulate owners of other "working" bibliographies to publish theirs. As noted below, this bibliography is online and may be accessed via the ARPANET.

Because this bibliography was incidental to other work rather than an end in itself, little time was spent in trying to make it comprehensive, or in supplying extensive annotations. It will also be noted that relatively little material has been added during the past year.

With respect to its subject, this bibliography was always intended to cover only a limited set of topics within the larger fields of operating system and computer security, specifically the following:

Operating system requirements, policies and mechanisms, for insuring user and operating system program and data integrity and confidentiality.

The best way to get a feel for the topics included is to scan the index (see below). The following peripheral topics have been specifically excluded (except where they occur together with the main subject above):

User identification and authentication

Encryption

Hardware reliability

Error detection methods

System generation and initialization

Human integrity (e.g., administration and operations staff)

Physical installation security

Communications security

Broader economic, social, and political issues

A reviewed document was also not included if the bulk of its subject matter, as a whole and by section, was other than that described above, even though it may have contained relevant fragments.

Gray areas exist in which the boundaries of the protection field are not well defined. Ultimately, probably no completely satisfactory definition of protection exists that stops short of including the entire constraintive aspect of programming and operating systems, i.e., of including policies and mechanisms for all kinds of unintended occurrences.

Selectivity has also been based on quality and significance. The attempt was made to exclude documents deemed not to be potentially useful for future reference personally, and which could not be recommended to students of protection. Exclusion of such entries from published bibliographies should be a professional ethic, even at the expense of exposing the often subjective judgments of their editors.

In general, a reviewed document has been included in this bibliography if it was judged to contribute original and potentially helpful observations, insights, ideas, or descriptions; or to express old ones in new and potentially helpful ways.

A document was rejected for any of the following "editorial" reasons:

It has been superseded by a revision, or its essence has been republished in a more accessible source.

It is totally obsolete or is of historical interest only.

Its editorial quality is inadequate.

As a result of the level of effort put into it and the restrictions applied, only 173 entries appear. Approximately 154 documents that qualified by topic (at least to some extent) were reviewed and rejected or included and later dropped. In addition, about 65 entries are currently on my list of unseen documents whose titles suggest relevance.

A unique feature of this bibliography is its index. An indication of the maturity of a field of research is the extent to which its technical terms are defined and used consistently. Access to definitions and usages is especially important to students in the field. Also, the technical terms used usually comprise a fair index to the content of a document. For these reasons, and to experiment with the actual usefulness of such an index, as each document was reviewed its key terms, concepts, and topics were noted as a set of keywords. Little attempt has been made to standardize the forms occurring in the index, or to insure consistency, for example by going back to revise the

keyword set of a document reviewed two or three years earlier. None of the massaging ordinarily needed to enhance the usefulness of an index has been applied to this one; like the bibliography itself, it is published in its current unfinished state.

This document corresponds to a file residing in the computer at ISI whose ARPANET hostname is ISIB, and which runs under the TENEX Operating System. The TENEX filename of this document is <REPORTS>PBIB.TXT; it may be accessed over the ARPANET via an FTP program.

Each entry in the bibliography has the following fields:

An identifier local to this bibliography.

Author's (authors') name(s) as appearing on the title page of the document.

Title of the document.

Source data fields: periodical issue, publishing agency, publisher's document identifier, date, pages, etc. In some cases an NTIS order number is also provided.

An annotation (usually), starting with the characters "[x" and ending with "x]", which attempts to summarize the most significant topics or features of the document in about one sentence.

Keywords or lists of keywords, enclosed by angle brackets "<" and ">" and separated (within lists) by semicolons. These can also occur as part of the annotation fields of an entry.

Fields of an entry are separated by double spaces, except that the local identifier and the author-name field are separated by a tab character. Entries themselves are separated by null lines.

The following abbreviations occur in the bibliography:

ACMnn Proceedings, ACM Conference 19nn

CACM Communications of the ACM

IBM74d Data Security and Data Processing, Vol. 4 Study Results:
Massachusetts Institute of Technology IBM G320-1374 74.6

ICRS75 Proceedings, International Conference on Reliable Software,
April 21-23, 1975

IWCA73 Proceedings, International Workshop on Computer Architecture,
Grenoble, June 26-28, 1973

IWPOS74 Proceedings, International Workshop on Protection in Operating
Systems, Rocquencourt, France, August 13-14, 1974. Institut de

Recherche d'Informatique et d'Automatique, BP 5 - Rocquencourt 78150
Le Chesnay, France

FJCCnn AFIPS Conf. Proceedings 19nn Fall Joint Computer Conference

NCCnn AFIPS Conf. Proceedings 19nn National Computer Conference

NTIS National Technical Information Service

PLOS73 Proc. of ACM SIGPLAN-SIGOPS Interface Meeting: Programming
Languages - Operating Systems, April 9-12, 1973

SJCCnn AFIPS Conf. Proceedings 19nn Spring Joint Computer Conference

SOSP75 Proc. of the Fifth Symposium on Operating Systems Principles,
November 19-21, 1975

BIBLIOGRAPHY

Abb+76 Abbott, R.P.; Chin, J.S.; Donnelley, J.E.; Konigsford, W.L.; Tokubo, S.; Webb, D.A. & Linden, T.A. (ed.) Security analysis and enhancements of computer operating systems. National Bureau of Standards Institute for Computer Sciences and Technology NBSIR 76-1041 76.4 68p [x Brief informal discussions of security flaws and enhancements; brief design overviews and descriptions of flaws found in IBM's OS/MVT, the UNIVAC 1100 series operating system, and the TENEX operating system for the DEC PDP-10. x] <security enhancements, types; errors, integrity, taxonomy/categories/examples>

AmbH77 Ambler, Allen L. & Hoch, Charles G. A study of protection in programming languages. SIGPLAN Notices 12,3(77.3) (Proc. ACM Conf. on Language Design for Reliable Software) 25-40 [x Compares protection features of Pascal, Concurrent Pascal, Euclid, CLU, and Gypsy, with respect to <abstract data types; modules; scope rules; parameter passing>, using the "prison mail system" problem as an example. x] <access control features, language; classes; protection principles, languages; selective access>

Ames74 Ames, Stanley Richard, Jr. File attributes and their relationship to computer security. Case Western Reserve Univ. Dept. of Computing and Information Sciences Report No. 1167 74.6 89p [x Successively refined models of a tree-structured file system to include security requirements governing observation and modification of file attributes. x] <security events/axioms; repositories; agents; security classes; observe relation; modify relation; clearance; executors; clusters; view relation; alter relation; manipulators; accumulations; looks-at relation; changes relation; mandatory/discretionary controls>

Ande72a Anderson, James P. Information security in a multi-user computer environment. Advances in Computers Vol. 12 Academic Press 1972 1-36 [x Includes a broad survey of problems, models, and mechanisms. x] <isolation mechanisms; two state operations; i/o channels; virtual machines; program identity; implied sharing; descriptors; classification, derived>

Ande72b Anderson, James P. Computer security technology planning study. ESD-TR-73-51, vol. II. 72.10 137p. [x Detailed research and development proposal, to include security models, design of a <security kernel>, and architecture, hardware, and systems studies. x] <reference monitor; descriptor-based architecture; vulnerabilities, classes; implied sharing vulnerability; scavenging problem; incomplete parameter checking; asynchronous interrupt vulnerability; asynchronous i/o vulnerability; trojan horse problem; file authorization; authorization, files; hierarchical access control; data management systems; aggregation; inference; procedural controls>

Andr74 Andrews, G.R. COPS--a mechanism for computer protection. IWPOS74 5-25 Also, Cornell Univ. Dept. of Computer Science CU-CSD-74-214 74.10 35p [x <Capability-based protection mechanism>. x] <actor, process/procedure; mechanism, def.; principle of control; manager problem; mutual suspicion; confined computation, def.; protection state; monitors; primitives, protection state/environment; information structures, control/computing; capabilities; attributes, access/control; environment; basic monitor; message confinement; protection, logical vs. physical>

Att+76 Attanasio, C.R.; Markstein, P.W. & Phillips, R.J. Penetrating an operating system: a study of VM/370 integrity. IBM Systems J. 15,1(1976) 102-116

Atta73 Attanasio, C.R. Virtual machines and data security. Proc. ACM SIGARCH-SIGOPS Workshop on Virtual Computer Systems, 73.3 ACM 1973 206-209 [x <Virtual machines, security advantages>. x] <intention, explicit>

Atta74 Attanasio, C.R. An additional protection ring for virtual machine systems. IBM Research RC 5617 (#22625) 74.11.14 12p [x Proposed method of allowing communication between virtual machines by extending each with a virtual control store (VCS) and virtual microprocessor, protected from the user. x]

Bar+67 Barron, D.W.; Fraser, A.G.; Hartley, D.F.; Landy, B. & Needham, R.M. File handling at Cambridge University. SJCC67 163-167 [x Early <file access control scheme>. x] <part owner>

Baum75 Baum, I.R. The architectural design of a secure data base management system. (Thesis) Ohio State Univ. Computer and Information Science Research Center 1975

BeIL73a Bell, D.E. & LaPadula, L.J. Secure computer systems: mathematical foundations. Mitre Corp. MTR-2547, vol. I ESD-TR-73-278, vol. I 73.11 37p NTIS: AD-770 768 [x Formal model defined from the viewpoint of general systems theory. x] <security, def.; classifications; need-to-know categories; access attributes; request sequences; decision sequences; access matrices; state sequences>

BeIL73b Bell, D.E. & LaPadula, L.J. Secure computer systems: a mathematical model. Mitre Corp. MTR-2547, vol II ESD-TR-73-278, vol. II 73.11 58p NTIS: AD-771 543 [x Enhancement of the model presented in [BeIL73a]; <privacy restrictions> are shown to be enforced under certain state transformations. x] <access attributes; access matrix; control access; error decision; question decision; security condition; x-property; security principle; interactivity principle; tranquility principle; classifications; categories (access privileges); request sequences; decision sequences; state sequences; rule, def.; rules of operation>

Bell74 Bell, D.E. Secure computer systems: a refinement of the mathematical model. Mitre Corp. ESD-TR-73-278, Vol. III MTR-2547, Vol. III 74.5

BelL76 Bell, D.E. & LaPadula, L.J. Secure computer system: unified exposition and Multics interpretation. Mitre Corp. MTR-2997 ESD-TR-75-306 76.3 129p

BelW74 Belady, L.A. & Weissman, C. Experiments with secure resource sharing for virtual machines. IWPOS74 27-33 [x Results and methodology of VM/360 evaluation. x] <isolation; sharing, information vs. other resources; channel programs, vulnerabilities; asynchronous modification of instructions; security, def.>

Bing65 Bingham, Harvey W. Security techniques for EDP of multilevel classified information. Burroughs Corp. 4424-65-112 65.12 186p NTIS: AD-476 557 [x Includes broad treatment of early hardware and software mechanisms. x] <user control profile; program reference table; file control code>

Bis+75 Bisbey, Richard, II; Popek, Gerald & Carlstedt, Jim Protection errors in operating systems: inconsistency of a single data value over time. Univ. of Southern California Information Sciences Institute ISI/SR-75-4 75.12 16p [x Type of protection error in which the value of a variable can be changed between two operations for which it is assumed to remain constant. x] <consistency, single variables; time-of-check-to-time-of-use error; parameter passing; errors, examples>

Bran73 Branstad, Dennis K. Privacy and protection in operating systems. Computer 6,1(73.1) 43-46 Also, Operating System Review 7,1(73.1) 9-17 [x Report of Workshop on Privacy and Protection in Operating Systems, Princeton, NJ, June 12-14, 1972. x] <generic weaknesses; residue; parameter checking, incomplete; asynchronous i/o; trojan horse attack; covert channels>

Brat75 Bratt, Richard Glenn Minimizing the naming facilities requiring protection in a computing utility. MIT Project MAC TR-156 Also, Honeywell Information Systems Federal Systems Operations ESD-TR-76-161 75.9 129p [x Simplification of the Multics security kernel by removing <reference name space> management and directory pathname resolution. x] <segment as unit of protection; descriptors; security kernel, name management functions; directory initiation; detectability; directories, protection; rings, protection; access control lists; lying to prevent detectability>

BroS71 Browne, Peter S. & Steinauer, Dennis D. A model for access control. 1971 ACM SIGFIDET Workshop: Data Description, Access and Control 241-262 [x Model for enforcing privacy restrictions on the basis of object <classification> and category, active object <authority>, and <access lists>. x] <protection groups; access clique; authorization; access control; categories; high water mark; objects; contextual sensitivity; privilege list>

Brow71 Browne, Peter S. Data privacy and integrity: an overview. 1971 ACM SIGFIDET Workshop: Data Description, Access and Control 237-240 [x Basic distinctions. x] <privacy; integrity, data; input validation;

isolation; identification; authorization>

Burk74 Burke, Edmund L. Synthesis of a software security system. ACM74 648-658 [x Software verification schema based on four <levels of representation>--mathematical model, <formal specifications>, algorithms, and machine language; sketch of its application to the MITRE security model and a <security kernel> for the PDP 11/45. x] <reference monitor>

Car+71 Carroll, John M.; Martin, Robert; McHardy, Lorine & Moravec, Hans Multi-dimensional security program for a generalized information retrieval system. FJCC71 571-577 [x Scheme for controlling access at the file, record and item levels. x] <item key; item code; record key; record code; data base scheme; record level control; field level control>

Carl76 Carlstedt, Jim Protection errors in operating systems: validation of critical conditions. Univ. of Southern California Information Sciences Institute ISI/SR-76-5 76.5 33p [x Type of protection error in which insufficient validation exists to insure that an assumed condition holds. x] <validation errors; errors, examples; validity vs. integrity; criticality; influencability/influentiality>

Cha+75 Chamberlin, D.D.; Gray, J.N. & Traiger, I.L. Views, authorization, and locking in a relational data base system. NCC75 425-430 [x "Views prescribe what can be seen. Authorization prescribes what can be done to what is seen. Locks are a dynamic kind of authorization which prescribe what can be done to what is seen at this instant." x] <views, data base; authorization types, data base; grant; revocation; access control, data base; value dependent authorization>

ChaS76 Chandersekaran, C.S. & Shankar, K.S. On virtual machine integrity. (Letter) IBM Systems Journal 15,3(76) 264-269 [x Criticism of the imprecise terminology and the conclusion of [DonM75]. x] <hierarchically structured systems; security, def.; privacy, def.; integrity, def.; mechanism requirements; isolation, def.; security kernel>

Clar75 Clark, David D. & Redell, David D. Protection of information in computer systems. Tutorial, Comcon 75 Fall IEEE Computer Society 1975 260p

CohJ75 Cohen, Ellis & Jefferson, David Protection in the Hydra Operating System. Operating Systems Review 9,5(1975) (SOSP75) 141-160 [x Description of the Hydra protection mechanisms and their application to several well-known <protection problems>. x] <protection, access/control; decisions, prior/future; decisions, unilateral/negotiated; procedural embedding; capabilities; ownership; subsystems; amplification; rights, generic/auxiliary; C-list; templates; domains; seal/unseal; rights, required/new; mutual suspicion; modification; propagation; conservation; confinement; initialization; revocation, immediate/permanent/selective/partial/temporal/sharing; freezing; accounting; lost objects; aliases>

Con+72a Conway, R.W.; Maxwell, W.L. & Morgan, H.L. On the implementation of security measures in information systems. CACM 15,4(72.4) 211-220

[* How access control decisions based on <static privacy conditions> can be made more efficiently at compile time. *] <access matrix; virtual user; data dependent conditions; compile time access control; security vs. privacy; decision rule; column system; diagonal system; compile-time checking>

Con+72b Conway, R.W.; Maxwell, W.L. & Morgan, H.L. Selective security capabilities in ASAP--a file management system. SJCC72 1181-1185
[* <Data base scheme> incorporating both <record level control> via <access control lists> and <field level control> via a <lock-and-key mechanism>. *] <directory of authorized users; security class, field; compile-time checking>

Coss72 Cosserat, D.C. A capability oriented multi-processor system for real-time applications. Proc. First Intern. Conf. on Computer Communication. IEEE Computer Society or ACM 282-289 [* Includes a brief description of the <capability architecture> of the Plessey System 250. *] <protected subroutines>

Coss74 Cosserat, D.C. A data model based on the capability protection mechanism. IWPOS74 35-53 [* Capabilities are treated as protected data structure pointers that can be freely copied. *] <capability segments; segment as unit of protection; control, transfers of; capabilities vs. pointers>

DalD74 Daley, Robert C. & Donohue, James P. Security and authorization --semantics and examples. IBM74d 135-149 [* Suggestions for improving IBM's Resource Security System, among other things by allowing disseminated control of authorization and providing for <protected subsystems>, including <execute-only access> to programs. *] <security officer; user groups; file groups; authorization classes; access types>

DalN65 Daley, R.C. & Neumann, P.G. A general-purpose file system for secondary storage. FJCC65 213-229 [* Early description of the Multics file system. *] <hierarchical access control; file directories; access paths; usage attributes; access control lists; trap list>

Den+74 Denning, D.E.; Denning, P.J. & Graham, G.S. Selectively confined subsystems. IWPOS74 55-61 [* Implementation of a <confinement mechanism> satisfying several necessary properties, and its insufficiency. *] <confinement problem; error conditions; protected subsystems; selective confinement; engagement; mutual exclusion, customer processes; closure, confidentiality; nonleakage; transitivity of engagement; declassification; disengagement; nonretention; covert channels; compile-time checking>

DenD77 Denning, Dorothy E. & Denning, Peter J. Certification of programs for secure information flow. CACM 20,7(77.7) 504-513 [* <Lattice model> of <information flow>, and <certification mechanism> based on it that verifies the security of information flow in a program, including the <confinement property>. *] <security classes; flow relation/policies; property lattice>

Denn64 Dennis, J.B. Program structure in a multi-access computer. MIT Project MAC TR-11 1964 [x Introduces the notion of <spheres of protection>. x]

Denn65 Dennis, Jack B. Segmentation and the design of multiprogrammed computer systems. J. of the ACM 12,4(65.10) 589-602 [x "...Hence it is preferable that a protection mechanism operate in the name space of a computation..." x] <segment as unit of protection; spheres of protection>

Denn75 Denning, Dorothy E. Secure information flow in computer systems. (Thesis) Purdue Univ. Dept. of Computer Science CSD TR-145 75.5 177p

Denn76a Denning, Dorothy E. A lattice model of secure information flow. CACM 19,5(76.5) 236-243 [x Formulation of information flow security requirements in terms of lattice properties; run-time and compile-time enforcement mechanisms for static and dynamic security environments. x] <information flow, explicit/implicit; security classes; security clearances; binding, static/dynamic; data mark machine; compile-time/run-time enforcement>

Denn76b Denning, Peter J. Fault tolerant operating systems. ACM Computing Surveys 8,4(76.12) 359-389 [x Broad tutorial and survey: how <capability architectures> facilitate the techniques required for <confinement of errors>. x] <process isolation; environment, open/closed; capability list; access code; name, local/system; access list; storage capability; enter capability; multiple domain processes; protected entry; domain changing; attenuation of privilege; privilege state mechanism; privilege number; descriptors, privacy; virtual machines; resource control; decision verification; encapsulation, process>

DenV66 Dennis, Jack B. & Van Horn, Earl C. Programming semantics for multiprogrammed computations. CACM 9,3(66.3) 143-155 [x Introduces the notion of <capability list>. x] <spheres of protection; c-list; segment as unit of protection>

DonM75 Donovan, J.J. & Madnick, S.E. Hierarchical approach to computer system integrity. IBM Systems J. 14,2(1975) 188-202 [x Argues probabilistically that a <hierarchically structured operating system> on a <virtual machine monitor> can be more secure than a conventional multiprogramming system. x] <integrity, def.; security, def.; redundant security>

Down73 Downey, Peter J. Secure military computing systems. In Schell, Roger R.; Downey, Peter J.; and Popek, Gerald J. Preliminary Notes on the Design of Secure Military Computer Systems. Air Force Systems Command Electronic Systems Division Directorate of Information Systems Technology MCI-73-1 73.1 47p [x <Military security model> focusing on the structure of <access functions> and of a <security kernel>. x] <objects; subjects; access attributes (types); access relation; access monitor; security state; update monitor; constraints, updation (policy); consistency, policy; access control lists; locks and keys; compartments; need to know; user responsibility; control attribute; owner attribute;

accesses, normal vs. security update; breaches, internal vs. external>

Ell74 Ellis, Clarence A. Analysis of some abstract measures of protection in computer systems. Univ. of Colo. Department of Computer Science. CU-CS-043-74 74.5 46p NTIS: PB-235 297 [x Defines absolute, relative, and minimum <degree of protection> in terms of assignments of <access codes> to subjects and objects, gives assignments that maximize these, and applies this to hierarchical systems. x] <isolation level>

Eng172 England, D.M. Architectural features of System 250. Infotech State of the Art Report on Operating Systems. 1972 12p [x Description of the <capability architecture> of the Plessey System 250. x] <privileged mode, nonnecessity; capability registers; enter access>

Eng174 England, D.M. Capability concept mechanism and structure in System 250. IWPOS74 63-82 [x "...everything...is reducible to data structures...built up from...an interconnected network of capabilities..." x] <capability architecture; capability registers, real/virtual; load/store, capabilities; central capability segment; enter capability; privileged mode, nonnecessity; resources as protected data structures>

Eva167 Evans, David C. & Leclerc, Jean Yves Address mapping and the control of access in an interactive computer. SJCC67 23-30 [x Proposal for integrating access control into the (virtual) addressing and parameter-binding mechanisms of an operating system. x] <parameter spaces; entry, protected; access path control; address map, control information>

Fabr68 Fabry, R.S. Preliminary description of a supervisor for a machine oriented around capabilities. Univ. of Chicago Institute of Computer Research Quarterly Progress Report No. 18, I-B 1-97 68.8 [x Sketch of design and implementation. x] <supervisor as interaction controller; capabilities; capability registers; segment as unit of protection; virtual machines; capability segments; access codes; capabilities, user; enter access>

Fabr71 Fabry, R.S. List-structured addressing. (Thesis) Univ. of Chicago 1971

Fabr73 Fabry, R.S. Dynamic verification of operating system decisions. CACM 16,11(73.11) 659-668 [x An independent consistency check for every decision involving process interactions. x] <isolation; message system>

Fabr74 Fabry, R.S. Capability-based addressing. CACM 17,7(74.7) 403-412 [x Rationale, comparison, and implementation considerations of schemes in which addresses are capabilities containing unique segment identifiers. x] <capability-based addressing; capabilities, representation integrity, approaches; capabilities and storage allocation; enter access/instruction>

Fent73 Fenton, J.S. Information protection systems. (Thesis) Cambridge

Univ. Computer Laboratory 1973

Fent74 Fenton, J. S. Memoryless subsystems. Computer J. 17,2(74.5) 143-147

Fer+74 Ferrie, J.; Kaiser, C.; Lanciaux, D. & Martin, B. An extensible structure for protected systems' design. IWPOS74 83-105 [x Object and type construction/destruction and call/return operations in a <capability-based system, extensible>. x] <agents; objects; mutual suspicion; transition type; descriptor (capability); encapsulated type; case/uncase operators; domains; environment of domain; operators, strong/weak; kernel; type transformation; call/return operations, domain>

Fer+75 Fernandez, E.B.; Summers, R.C. & Coleman, C.D. An authorization model for a shared data base. Proc. 1975 SIGMOD Intern. Conf. 9p [x Access matrix is extended to allow entries to contain predicates that depend on any data in the data base; most checking is done at compile time. x] <data base access control; access matrix, extended; access predicates; allocate (access type); functional access; administer (access type); inherit (access type); use (access type); compile-time checking>

FerS76 Fernandez, Eduardo B. & Summers, Rita C. Integrity aspects of a shared data base. NCC76 819-827

Frie70 Friedman, T.D. The authorization problem in shared files. IBM Systems J. 9,4(1970) 258-280 [x Broad treatment of access control requirements and considerations, recommending <compartmentalization> and data element <tagging>. x] <integrity, system; authorization, def.; privilege structure, complexity; names, concealment; passwords, file; privileges, program/user; field, protected; isolation, authorization mechanism; single-tag rule; group (protection category); levels of classification, undesirability; authority hierarchies, undesirability; clique, user; matrix, user-privilege; user profile>

Gain72 Gaines, R. Stockton An operating system based on the concept of a supervisory computer. CACM 15,3(72.03) 150-156 [x <File access control> via a <lock-and-key scheme>. x] <protection categories, file/process, read/write>

GeoS73 George, James E. & Sager, Gary R. Variables--bindings and protection. SIGPLAN Notices 8,12(73.12) 18-29 [x Six "mini-languages" differing in their schemes for controlling the scope and accessibility of names. x] <names, scope/sharing/protection>

Gla+75 Gladney, H.M.; Worley, E.L. & Myers, J.J. An access control mechanism for computing resources. IBM Systems J. 14,3(1975) 212-228

Grah68 Graham, Robert M. Protection in an information processing utility. CACM 11,5(68.5) 365-369 [x Protection requirements, and a model featuring <rings of protection>. x] <privileged instructions, deficiency; need-to-know principle; two-mode systems, deficiency; layers of protection, advantages; segment as unit of protection; access bracket; call bracket;

ring bracket; user descriptor; gatekeeper; calls, cross-ring; asynchronous modification of arguments>

GraD72 Graham, G.Scott & Denning, Peter J. Protection--principles and practice. SJCC72 417-429 [x <Access matrix model>: enforcement rules, protection commands, applicability, and implementation considerations. x] <levels of protection; mutually suspicious subsystems; memoryless subsystems; completeness requirement; objects; subjects; access rules, requirements; protection state; access attributes; monitor, object type; identification, subjects; update rules, access matrix; copy flag; owner attribute; control attribute; transfer command; grant command; copy flag, transfer-only; domains; hierarchies, subject, advantages; universal subject; untrustworthy subsystems; indirect access; debugging problem; trust; trustworthy subjects; revocability; c-list; access control list; authority list; lock list; key>

Grah71 Graham, G.Scott Protection structures in operating systems. (Masters Thesis) Univ. of Toronto Department of Computer Science 71.8

GriW76 Griffiths, Patricia P. & Wade, Bradford W. An authorization mechanism for a relational data base system. IBM Research RJ 1721 (#25154) 76.2.11 32p [x Mechanism providing for restricted sharing of data bases by allowing the granting and revocation of privileges and the definition and sharing of <views>. x] <data base privacy mechanisms; revocation, recursive; grants, labeled>

Har+75 Harrison, Michael A.; Ruzzo, Walter L. & Ullman, Jeffrey D. Protection in operating systems. CACM 19,8(76.8) 461-471 [x <Safety> of a protection system modelled by a given general <access matrix model> is undecidable. x] <rights, generic; configuration, protection system; own right; leaking, rights; undecidability, safety>

HarH75 Hartson, H.R. & Hsiao, D.K. Languages for specifying protection requirements in data base systems. Part I. Ohio State Univ. Computer and Information Sciences Research Center OSU-CISRC-TR-74-10 75.1 67p NTIS: AD/A-006 280 [x Models the <authorization process> and the <enforcement process> in a <security space> of users, authorizers, resources, operations, and value states. x] <access control model; authorization specification, validity/consistency; access condition; protection specification language; access matrix model, limitations; ownership, absolute/conditional; system administrator; withdrawal of rights; subownership; memoryless specifications; access history; authorized subspace; dominance, access history; access decision timing; domain of authorization; auxiliary invocation, domain; validation of inputs; progressive authorization; amplification; extended resources; franchise, user/operation/resource unit; partial rejection>

Harr75 Harrison, Michael A. On models of protection in operating systems. Lecture Notes in Computer Science. Vol 32: Mathematical Foundations of Computer Science 1975 (J. Becvar, ed.) Springer-Verlag 1975 46-60

Hart75 Harston, R. Rex Languages for specifying protection requirements

in data base systems--a semantic model. Ohio State Univ. Computer and Information Science Research Center CISRC-TR-75-6 75.8 248p NTIS: AD-018 284/0GA

Hel+75 Held, G.D.; Stonebraker, M.R. & Wong, E. INGRES--a relational data base system. NCC75 403-416 [* Access is controlled and integrity assured by means of <query modification>. *] <access control, data base; integrity, data base; restrict statement; integrity constraint>

HinS75 Hinke, T.H. & Schaefer, Marvin Secure data management system. System Development Corp. TM-(L)-5407/007/00 RADC-TR-75-266 75.11 197p NTIS: AD-A019 201

Hoff71 Hoffman, Lance J. The formulary model for flexible privacy and access controls. FJCC71 587-601 [* Proposes a <formulary> procedure to intercept all accesses to the data element to which it is attached and to determine the right of access, possibly via a dialogue with the user. *] <control profile, user; authority item; data dependent decisions>

HolB76 Hollingworth, Dennis & Bisbey, Richard, II Protection errors in operating systems: <allocation/deallocation residuals>. Univ. of Southern California Information Sciences Institute ISI/SR-76-7 76.6 17p [* Type of protection error in which a residual from a previous allocation remains accessible after a new allocation. *] <residual, content/access; errors, examples>

Hsi+74 Hsiao, D.K.; Kerr, D.S. & McCauley, E.J., III A model for data secure systems (Part I). Ohio State Univ. Computer and Information Science Research Center OSU-CISRC-TR-73-8 74.2 44p [* General protection specifications can be used to define users' views (and accessibility domains) of a data base, since they have the same form as retrieval specifications (Boolean functions). *]

Hsia68 Hsiao, David K. A file system for a problem solving facility. (Thesis) University of Pennsylvania The Moore School of Electrical Engineering Report No. 68-33 68.5 157p NTIS: AD-671 826 [* Both <file access control> and <record level control>, based on information in the <authority item> associated with each user. *] <login program, file; owner; authentication, file login program>

HsiM74 Hsiao, D.K. & McCauley, E.J., III A model for data secure systems. (Part II) Ohio State Univ. Computer & Information Science Research Center OSU-CISRC-TR-74-7 74.10 45p

IBMx71 (anon) OS/MVT with resource security: general information and planning manual. IBM GH20-1058-0 71.12 28p [* Description of Resource Security System features. *] <security levels; access category; need to know; security officer; definition of controlled resources; grouping of users and resources; authorization; program-restricted data sets; associative programs>

Jans74 Janson, Philippe Arnaud Removing the dynamic linker from the

security kernel of a computing utility. (Thesis) MIT Project MAC TR-132 74.6 128p [x Protection implications and design of a <dynamic linker> external to the security kernel, in terms of an abstract storage, name space, and protection model; implementation in Multics. x] <principle of least privilege; security kernel, criteria; object; subject; capability; gate; protected subsystem; name space model; system initialization; link fault handling; static storage allocation; rings, protection>

Jone73 Jones, Anita K. Protection in programmed systems. (Thesis) Carnegie-Mellon Univ. Department of Computer Science 73.6 145p NTIS: AD-765 535 [x Model of a protection mechanism based on <environments>: collections of <constituent rights>, associated with objects or their components; <declared rights>, associated with procedures; and <dynamic rights>, associated with procedure invocations. x] <environment crossing operators; protection, def.; object; access; type; enforcement rule; right transfer rule/primitives; capability; ownership; access matrix; environment binding rule; environment representations; checking, dynamic vs. static; parameter passing; amplification; protection state, transformations, closure; memoryless procedure; dynamic type creation; subsystem; suitability factor; need-to-know principle; accuracy measure; protection systems, comparison; policy vs. mechanism>

JonL75 Jones, Anita K. & Lipton, Richard J. The enforcement of security policies for computation. Operating Systems Review 9,5(75) (SOSP75) 197-206 [x For any viewing program and any security policy there exists a sound and <maximally complete mechanism>, but which cannot always be constructed; construction of sound <surveillance mechanisms>. x] <v functions; o operators; high water mark; policy, def.; mechanism, def.; soundness; completeness; confinement; memoryless subsystems; data security; observability postulate; information control vs. access control>

JonL76a Jones, Anita & Lipton, Richard J. (Letter) Operating Systems Review 10,2(76.4) 7-8 [x Reply to [Rote76] on the <soundness> of the privacy restriction mechanism at issue. x]

JonL76b Jones, Anita K. & Liskov, Barbara H. A language extension for controlled access to shared data. IEEE Trans. on Software Engineering SE-2,4(76.12) 277-285 [x Rules enforceable at compile time governing the binding of <capabilities (variables)> to strongly typed, structured objects, and implementing a wide class of controlled sharing policy. x] <compile-time checking; type-module; qualified type; binding rules; amplification>

JonW75 Jones, Anita K. & Wulf, William A. Towards the design of secure systems. Software--Practice and Experience 5,4(75.10-12) 321-336 [x Distinction between policies and mechanisms; capability-based mechanisms; the Hydra mechanism; types and examples of policies that can be implemented in Hydra. x] <objects; security policy, def.; appropriateness, policy; requirements, protection mechanisms; data base protection; capabilities, extended; mechanisms, functions; amplification; local name space; rights, kernel/auxiliary; capability transfer operations; environment crossing operations; checkrights; amplifyrights; template;

courier policy; monitoring policy; revocation policy; confinement policy; filtering policy; gatekeeper>

KamU77 Kam, John B. & Ullman, John D. A model of statistical databases and their security. ACM Trans. on Database Systems 77.3

KarS74 Karger, Paul A. & Schell, Roger R. MULTICS security evaluation: vulnerability analysis. USAF Electronic Systems Division ESD-TR-74-193, Vol. 11 74.6 155p [x Overview of Multics security controls; detailed descriptions of several vulnerabilities identified, confirmed, and exploited during an early 1973 analysis. x] <multi-level security; master mode; access control lists; rings, protection; subverter program; vulnerabilities, descriptions; validation of arguments, insufficient; penetration techniques; trap doors, classes; procedural vulnerabilities; utility programs, dump/patch; compiler trap doors; gatekeeper>

KohG75 Kohout, L. & Gaines, B.R. The logic of protection. Lecture Notes in Computer Science, Vol. 34 GI--5. Jahrestagung Springer-Verlag 1975 736-751

Lack74 Lackey, R.D. Penetration of computer systems. An overview. Honeywell Computer J. 8,2(74) 81-85 [x Categories of <penetration techniques> and some common <error types>. x]

Lam+77 Lampson, B.; Needham, R.; Randall, R. & Schroeder, M. Protection, security, reliability. Operating Systems Review 11,1(77.1) 12-14 [x Things to be done; misconceived problems. x] <absolute vs. defensive protection; numerical measure of security>

Lamp67 Lampson, Butler Wright Scheduling and protection in an interactive multi-processor system. (Thesis) Univ. of Calif., Berkeley 67.3 82p [x <Control protection> via a per-process vector of allowable instructions; <memory protection> implications of various addressing schemes. x]

Lamp68 Lampson, Butler W. A scheduling philosophy for multiprocessing systems. CACM 11,5(68.5) 347-360. [x Includes the scheme for <control protection> presented in [Lamp67]. x]

Lamp69 Lampson, B.W. Dynamic protection structures. FJCC69 27-38 [x Detailed treatment of <capabilities> and <domains>, especially with respect to problems of <control transfers> and sharing. x] <access key; rings; gates; proprietary programs; passwords, file; access control lists>

Lamp71 Lampson, Butler W. Protection. Operating Systems Review 8,1(74.1) 18-24 [x Use and possible implementations of the <access matrix>. x] <domain; message system; identification; object system; access attributes; copy flag; revocation; ownership; capability list; access key; access lock list; access control procedure, per object>

Lamp73 Lampson, Butler W. A note on the <confinement problem>. CACM 16,10(73.10) 613-615 [x The problem of preventing a <service program>

from leaking information. *} <covert channels; memorylessness; transitivity of confinement; masking, channels>

Lamp74 Lampson, Butler W. Redundancy and robustness in memory protection. Proc. IFIP Congress 74, Vol. 1 North-Holland/American Elsevier 1974

128-132 [* Approaches to protection as the control of efficient communication between memory-sharing domains, as opposed to protection as isolation of message-sending/receiving domains. *} <domain; chattels; protection, def.; protection, absolute/defensive; isolation; message model; memory environment; scope of domain; communication by value vs. pointer; authentication; trademarks; registrar; memory protection; sharing, slow/fast; capabilities>

LamS75 Lampson, Butler W. & Sturgis, Howard E. Reflections on an operating system design. CACM 19,5(76.5) 251-265 [* Strengths and weaknesses of the capability-based Cal operating system in retrospect. *} <capabilities; domains; c-lists; seals; access keys>

Laue74 Lauer, H.C. Protection and hierarchical addressing structures. IWPOS74 137-148 [* Advantages/disadvantages and protection implications of <nested address space systems> versus <global object name systems>. *} <scope of names>

Less68 Lesser, V.R. A multi-level computer organization designed to separate data-accessing from computation. Stanford Univ. Computer Science Department Tech. Rep. CS90 68.3

Lind73 Lindsay, Bruce Suggestions for an extensible capability-based architecture. IWCA73 20p [* Possibilities for the representation, protection, passing, creation, interpretation, & encapsulation/retrieval of <capabilities>.*]

Lind74 Linden, T.A. Different goals for protection. IWPOS74 149-153 [* A protection mechanism should (1) provide for rigorous data security, (2) facilitate the construction of reliable software, (3) support the implementation of special protection mechanisms, and (4) aid in guaranteeing the protection mechanism's own integrity and correctness. *} <goals, protection mechanisms>

Lind76a Linden, Theodore A. Protection: a nuisance or an opportunity? Digest of Papers, COMPCON Fall 76 IEEE 30-35

Lind76b Linden, Theodore A. Operating system structures to support security and reliable software. ACM Computing Surveys 8,4(76.12) 409-445 [* Tutorial on small protection domains implemented via <capability-based addressing>; <extended-type objects>; and the applicability of these to reliable software and system security. *} <security, def.; subjects; objects; access modes; access right; domains; capabilities; access matrix model; domain switching; protected procedure; enter right; mutual suspicion; principle of least privilege; defensive programming; Trojan horse problem; intermediaries; directories; revocation; amplification; indirection; extended-type manager; modularity>

discretionary/nondiscretionary controls; classification systems>

Lipn72 Lipner, Steven B. Computer security research and development requirements. The MITRE Corporation MTR-142 73.2 <reference monitor; need to know; segment, basis for access control>

Lipn75 Lipner, Steven B. A comment on the confinement problem. Operating Systems Review 9,5(1975) (SOSP75) 192-196 [x Formalization of confinement requirements in terms of the <x-property> and an approach to proving that this property hold for all <storage channels> and <legitimate channels> of an operating system; comment on the use of <virtual time> to eliminate <covert channels>. x] <confinement problem; hiding, information; high water mark; o-functions; v-functions>

LisZ74 Liskov, Barbara & Zilles, Stephen Programming with <abstract data types>. SIGPLAN Notices 9,4(74.4) (Proc. Symp. Very High Level Languages) 50-59 [x A form of <encapsulation> is provided by defining classes of objects in terms of the <cluster of operations> available on them. x] <type checking>

Mano71 Manola, Frank A. An extended data management facility for a general-purpose time sharing system. (Master's thesis) Univ. of Pennsylvania 71.05 NTDS: AD-724 801 160p [x Extensions to scheme of [Hsia68]: <deny/allow descriptions> for <record level control>; <field level control> by deleting restricted keywords and corresponding fields from accessed records. x]

ManW75 Manola, Frank A. & Wilson, Stanley H. Data security implications of an extended subschema concept. Naval Research Laboratory Report 7905 75.7.15 16p

Mart73 Martin, James Security, accuracy, and privacy in computer systems. Prentice-Hall 1973 640p [x Includes descriptions of a variety of authorization schemes, including IBM's Resource Security System. x] <data security, def.; authorization structures; stratification; compartmentalization (isolation); authorization tables, user/data; transaction types; categories, user/data; locks, data record; passwords, file; capability (training); lockwords; zones, data; authorization level; security levels; access categories; need to know; integrity, system; security officer; program restricted data; security by association>

Mcph74 McPhee, W.S. Operating system integrity in OS/VS2. IBM Systems J. 13,3(74) 230-252 [x Seven types of <system integrity> errors in forerunner systems, and their solutions in OS/VS2. x] <time-of-check-to-time-of-use problem; validity checking; identification, objects; storage protection; restricted names; authorized program facility; program authorization; user-supplied addresses; serialization mechanisms; user data passed as system data; errors, types/examples>

Mill75 Millen, Jonathan K. Security kernel validation in practice. CACM 19,5(76.5) 244-250 [x Detailed description of method and techniques used to prove security properties of a kernel for the PDP-11/45. x]

<security kernel; *-property; simple security condition; discretionary access control matrix; tranquility principle; indirect information paths; O-functions; V-functions; elimination rules; channels, legitimate/covert; time problems>

Mins76a Minsky, Naftaly Intentional resolution of privacy protection in database systems. CACM 19,3(76.3) 148-159 [* Operations on data retrieved from a data base by a user program (assumed written in a strongly typed language) can be restricted more strongly than with access control alone, by requiring the program to interact via a subschema containing <application rules> augmenting those of the given language. *] <intentional resolution; privacy (security); access control limitations; database protection; database subschema; right (part of application rule); brands; data analysis, prevention; intermediate results, hiding; confinement; confidence modules>

Mins76b Minsky, N. An activator based protection mechanism. Rutgers Univ. Dept. of Computer Science Technical Report No. 25 76.6 40p

MooC74 Moore, Charles G., III & Conway, Richard Program predictability and data security. Cornell Univ. Dept. of Computer Science TR 74-212 74.9 12p [* Capabilities can be controlled at the source language level, allowing identification of potential <indirect access> via analysis of the <information flow graph> as well as some <compile-time enforcement>, provided that the name interpretation and accessing semantics of that language are correctly defined and implemented.*] <security matrix>

Moor73 Moore, B.J. A classification of central processor architecture. IWCA73 19p [* Basic architectural concepts and the relationships between them, including processes, address spaces, and protection domains and mechanisms. *]

Moor74 Moore, Charles G., III Potential capabilities in Algol-like programs. Cornell Univ. Dept. of Computer Science TR 74-211 74.9 19p [* Potential flow of information in either direction between a given uninterpreted block B and a given variable may be determined from the <information flow graph> (for which the construction algorithm is given) of the program in which B is contained. *] <path condition; flow condition; capabilities, potential>

Morr73a Morris, James H., Jr. Protection in programming languages. CACM 16,1(73.01) 15-21 [* Proposes <seals> and <trademarks> for maintaining privacy and integrity of program module data. *]

Morr73b Morris, James H., Jr. Types are not sets. Conf. Record of ACM Symposium on Principles of Programming Languages, 73.10.1-3 120-124 [* Introduction of <seal operators, opaque/transparent> to guarantee the integrity of operators on and representations of objects of dynamic types. *] <authentication (type checking); secrecy (of representations)>

Need72 Needham, R.M. Protection systems and protection implementations. FJCC72 571-578 [* Per process <capability segments>; addressing via

<indirection tables> in which segment names are bound. ✖]

Need73 Needham, R.M. Protection--a current research area in operating systems. International Computing Symposium 1973. North-Holland/American Elsevier Publishing Company 1974 123-126 [✖ Motivation for and description of the capability/domain approach. ✖] <mechanisms, def.; security of information, def.; Trojan horse problem; file access via given program; regime of protection; segment as unit of protection; capabilities; enter capability; asynchronous events>

NeeW74 Needham, R.M. & Walker, R.D.H. Protection and process management in the "CAP" computer. IWPOS74 155-160 [✖ Non-hierarchical protection in a hierarchically-structured system. ✖] <capability segment; indirection table; ENTER capability; protected procedure; hierarchical process structure>

Neu+76 Neumann, Peter G.; Feiertag, Richard J.; Levitt, Karl N. & Robinson, Lawrence Software development and proofs of multi-level security. Proc. 2nd Intern. Conf. on Software Engineering IEEE 1976 421-428

Neu+77 Neumann, Peter G.; Boyer, Robert S.; Feiertag, Richard J.; Levitt, Karl N. & Robinson, Lawrence A provably secure operating system. Stanford Research Institute Project 4332 Final Report 77.2.11 483p [✖ A five-stage design methodology for general-purpose operating systems, with assertions stated and proved in an assertion language common to all stages; and the design of a secure operating system achieved via that methodology, structured as a hierarchy of abstract machines. ✖] <alteration principle; detection principle; denial of service; leakage, information; capabilities; objects; access code; type manager; revocation, selective; lost-object problem; Trojan horse attacks; mediated access; mutually suspicious subjects; memoryless operation; military security classification; need to know; inference; confinement principle; ✖-property; security kernel; C-list; subjects; domains; rings; call and return mechanism; capability channels; revocable copy; distinguished entry>

Neum73 Neumann, Peter G. (reporter) Report of evening session on protection. SIGPLAN Notices 8,9(73.9) (PLOS73) p27 <hidden channels; move-without-reading; read-without-copying; user interface; principle of maximum security; protection as restriction>

Owen71a Owens, Richard C. Primary access control in large-scale time-shared decision systems. (M.S. Thesis) MIT Project MAC TR-89 71.07 91p

Owen71b Owens, R. Evaluation of access authorization characteristics of derived data sets. Proc. 1971 ACM-SIGFIDET Workshop on Data Description, Access and Control 263-278

Palm73 Palme, Jacob Protected program modules in Simula 67. Research Institute of National Defense, Stockholm, Sweden FOA P Report C 8372 73.7 26p NTIS: PB-224 776 [✖ Requirements for <inter-module protection>

in programming systems and languages; most of them satisfied by Simula 67. A <HIDDEN specification> is proposed to prevent external access to class attributes.*] <classes, Simula 67>

Palm74 Palme, Jacob Software security. Datamation 20,1(74.1) 51-55
[* Brief tutorial. *] <deduction problem; file keys; error patterns>

ParP74 Parnas, D.L. & Price, W.E. Using memory access control as the only protection mechanism. IWPOS74 177-181 [* Argues that a virtual memory that provides a separate address space for each process is the only protection mechanism needed for basic access limitation. *]
<virtual memory as protection mechanism; need-to-know principle>

Pope73a Popek, Gerald J. Access control models. (Thesis) Harvard Univ. Center for Research in Computing Technology 73.2 153p NTIS: AD-761 807 [* Model based on a <restriction graph> whose nodes are <security objects> and edges are <access paths>; an <access map> from <actors> to nodes is implemented as a map from actors to boolean <key variables> and a map from edges to boolean <lock functions> of these variables; implementation algorithms. *] <locks and keys; collusion; filtering property; access graph; access control list>

Pope73b Popek, Gerald J. Correctness in access control. ACM73 236-241
[* <Set-theoretic model> of a <security system>, with proof of correctness. *] <inference problem; statistical access problem; kernel>

Pope74 Popek, Gerald J. Protection structures. Computer 7,6(74.6) 22-33 [* Brief but comprehensive survey. *] <mutually suspicious subsystems; memoryless process; statistical access; inference; errors, examples; parameter checking; synchronization; higher level errors; hidden channels; security object; passive object; active object; protection data; update program; enforcement rule; access matrix; rights transfer primitives; capabilities; environment binding; domain; sphere of protection; c-list; revocation; locked boxes; access control list; rings; seal/unseal; data-dependent access decisions; formularies; principle of least privilege; grain of protection; security kernel; virtual machines; locks and keys; scope of names; redundant checking; penetration>

PopK74a Popek, Gerald J. & Kline, Charles S. A verifiable protection system. SIGPLAN Notices 10,6(75.6) (ICRS75) 294-304 [* <Security kernels>, <virtual machines>, and program verification; design considerations of the UCLA-VM System. *] <security sensitive instructions; security kernel, functions; input/output; data security, def.; security objects; access predicates; updater; capability faulting; interruptibility; privileged code, exclusion from user process; scheduling; file management kernel; existence, knowledge of; kernels, levels; levels of mechanism; direct access predicate; indirect access predicate; objecthood; Morse Code problem; vulnerabilities, multilevel>

PopK74b Popek, Gerald J. & Kline, Charles S. Verifiable secure operating system software. NCC74 145-151 [* Approach to providing verifiable security; the applicability of <security kernel> designs, virtual machine

work, and program verification techniques; the UCLA-VM system. *}
 <constraints; security objects; accessible sets; policy, def.; modularity;
 owner; virtual machines; security primitives; Morse Code problem; updater>

Pric73 Price, William Robert Implications of a virtual memory mechanism
 for implementing protection in a family of operating systems. (Thesis)
 Carnegie-Mellon Univ. Computer Science Dept. 73.6 251p NTIS:
 AD-766 292 {* Operating system design featuring local working subsets of
 virtual address spaces as domains of protection; Parnas specifications of
 the operating system modules; formal verifications. *}

Rede74 Redell, David D. Naming and protection in extendible operating
 systems. (Thesis) MIT Project MAC TR-140 74.11 166p {* Features
 and goals of typical capability systems, especially those related to
 <revocation> and <type extendibility>, and a proposed implementation
 for a system that handles both these problems using <sealed
 capabilities>. *} <base level (kernel); domains and processes;
 communication, interprocess/interdomain; lost object problem; capabilities,
 protection of; identifiers, unique; caretaker domain; process control as
 capability privilege; ownership privilege; capabilities, indirect, chains;
 amplification, authorization; operators, non-monadic; capabilities, sealed;
 capability systems, goals; subletting; revoker capability; revocability,
 revocable; trust, changing levels; locker capability; extender capability;
 parameters, revocable; directories in capability system;
 comprehensibility of protection mechanisms; propagation, knowledge>

RedF74 Redell, D.D. & Fabry, R.S. Selective <revocation of capabilities>.
 IWPOS74 197-209 {* Goals for revocation schemes; two domain-independent
 schemes based on the notions of <indirection and control>; base-level
 implementation. *} <revocability of revocability; copy of capability,
 literal/revocable>

Rhod75 Rhode, R. Secure multilevel virtual computer systems. Mitre Corp.
 MTR-2890 ESD-TR-74-370 75.2 33p NTIS: AD/A-007 059

Rob+75 Robinson, Lawrence; Levitt, Karl N.; Neumann, Peter G. & Saxena,
 Ashok R. On attaining reliable software for a secure operating system.
 SIGPLAN Notices 10,6(75.6) (ICRS75) 267-284 {* Design methodology for
 and features of an operating system, with security and concurrent
 verification as goals. *} <type manager; protection theorems; security
 problems, special; security, def.; security kernel; capabilities; access
 vector; capability manager>

Rote74 Rotenberg, Leo J. Making computers keep secrets. (Thesis) MIT
 Project MAC TR-115 74.2 393p {* Mechanisms designed to solve classes
 of problems associated with <proprietary services; privacy restrictions>;
 and <authority hierarchies>. *} <segments; access control lists;
 capabilities; capability lists; domains, postulates; protection, abstract
 formulation; message system; caretaker program; encapsulation; argument
 passing; name spaces; naming hierarchy; access control packet; revocation;
 ownership of domains; call-out; call-in; capabilities, passing of; hidden
 data; argument spying; blind service; benign domain; billing; accounting>

channel; proprietary services, maintenance; locksmith; call packet;>

Rote76 Rotenberg, Leo J. (Letter) Operating Systems Review 10,2(76.4) 5-6 [✧ On the <soundness> of the privacy restriction mechanism described in [Rote74]. ✧]

Saal77 Saal, Harry J. A hardware architecture for controlling information flow. IBM Research RC 6414 (#27675) 77.2.28 30p [✧ Proposes an enhanced <capability vector mechanism> that improves on traditional capability schemes, which are shown to be inconsistent with the <least privilege principle> and with <information flow requirements>. (Abstract) ✧]

Salt75 Saltzer, J.H. & Schroeder, M.D. The protection of information in computer systems. Proc. of the IEEE 63,9(75.9) 1278-1308 [✧ Tutorial survey: basic principles; <list-oriented mechanisms> vs. <ticket-oriented mechanisms>; <dynamic authorization>, <authority structures>, and <protected subsystems>. ✧] <access control list; capabilities; design principles; failsafe principle; separation of privilege; least privilege; psychological acceptability; descriptor; principal; segment; domain; revocation; propagation; review of access; access controller; protection group; self control; hierarchical control; prescript; discretionary controls; sensitivity levels; confinement; operator-type restrictions; encapsulation>

Salt73 Saltzer, J.H. Protection and the control of information sharing in Multics. CACM 17,7(74.7) 388-402 [✧ Detailed description of the protection mechanisms associated with the Multics file system and virtual memory, with design principles and insights. ✧] <access control list; segment descriptor; permission principle; least privilege principle; protected subsystem; principals; compartments; access modes, primitive; intent specification; trap extension; file backup; system administrator; directories; authority hierarchy; locksmith; propagation of protection specifications; gates; rings of protection; argument checking; i/o operations; channel programs; system initialization; storage residue>

Salt74 Saltzer, Jerome H. Ongoing research and development on information protection. Operating Systems Review 8,3(74.7) 8-24 [✧ Informal survey of work at about 30 sites. ✧]

Salt76 Saltzer, J.H. Technical possibilities and problems in protecting data in computer systems. In R. Dierstein, H. Fiedler, and A. Schulz, Datenschutz und Datensicherung. J.P. Bachem Verlag, Cologne, Germany 76.9 27-36 [✧ A brief survey of protection methods and problems. ✧] <policy, informality; access control lists vs. capabilities; access control, container vs. data; limited-use systems; protected subsystems; classification systems; information flow, control; confinement; sharing, arbitrary patterns; inference>

Scha75 Schaefer, Marvin Secure data management system preliminary mathematical model. System Development Corporation RADC-TR-74.352 75.2 42p NTIS: AD-A007 748 [✧ Description and indicated proof

of a model of a <data access monitor> for a militarily secure data management system to be implemented in the Multics operating system. *}
 <classification; clearance; category; compartment; need to know;
 *-property; domination; high water mark; ownership; delete access;
 classification basis, record/field>

Sch75 Schiller, W.L. The design and specification of a security kernel for the PDP-11/45. Mitre Corp. MTR-2934 ESD-TR-75-69 75.5 117p NTIS: AD-A011 712 [* Design approach (levels of abstraction) and details of process management, segment management, security functions, and i/o. *]

Schr72 Schroeder, Michael D. Cooperation of mutually suspicious subsystems in a computer utility. (Thesis) MIT Project MAC TR-104 72.9 167p [* Design of a processor to provide controlled passing of parameters in <cross-domain calls> within a single computation. *]
 <protection, def./objectives; domain; capability; protected subsystem; segment as unit of protection; gates; access, static/dynamic; access control lists>

Schr75 Schroeder, Michael D. Engineering a security kernel for Multics. Operating Systems Review 9,5(1975) (SOSP75) 25-32 [* Notion of <security kernel>; motivation, method and activities involved in simplifying the Multics security kernel to make it easier to verify. *]
 <least common mechanism; rings, protection>

SchS72 Schroeder, Michael D. & Saltzer, Jerome H. A hardware architecture for implementing protection rings. CACM 15,3(72.3) 157-170
 [* Description of the Multics ring structure. *] <segment as unit of protection; domains; gates; ring structure/crossing; ring brackets, read/write/execute>

SevI74 Sevcik, K.C. & Tsichritzis, D.C. Authorization and access control within overall system design. IWPOS74 211-224 [* How the kernel of the SUE operating system solves various protection problems. *] <capabilities, control of transfer; rights, quantitative; isolation/interaction; suspicion; confinement; sponsor; excessive use of resources; volume, foreign/local; revocation; indirect capabilities>

SevI75 Sevcik, K.C. & Tsichritzis, D.C. Operating system design with security as an objective. INFOR J. 13,2(75.6) 159-174

Smit74 Smith, Grant N. The state of practice of computer security. IBM74d 163-178 [* Protection mechanisms found in current systems. *]
 <integrity, data/system; privilege levels; subfile access control; user categories; access types; access control list; virtual memory; virtual machines; virtual I/O>

Spi474 Spier, Michael J.; Hastings, Thomas N. & Cutler, David N. A storage mapping technique for the implementation of protective domains. Software--Practice and Experience 4(74) 215-230 [* Domain is a gated <caretaker procedure> together with the data encapsulated with it; implementation of an <activation/incarnation mechanism>. *] <domain

(protected subsystem); kernel>

Spie73 Spier, Michael J. A model implementation for protective domains. Intern. J. Computer & Info. Sciences 2,3(73.9) 201-229 [x A domain architecture in which domains are represented by sets of descriptors specifying not only allowable primitive access modes but also the mapping of information structures (objects of processing) onto storage areas (objects of ownership). x]

Spie74 Spier, M.J. A system theoretic look at the complexity of access control mechanisms. IWPOS74 225-241 [x The <complexity of an access control mechanism> is measured in terms of inter-module connectivity. x] <isolation; memorylessness and complexity>

Srin72 Srinivasan, C.V. A framework for a theory of protection. Rutgers Univ. Dept. of Computer Science Technical Report #16 72.05

Stor75 Stork, D.F. Downgrading in a secure multilevel computer system: the formulary concept. Mitre Corp. MTR-2924 ESD-TR-75-62 75.59p NTIS: AD-A011 696

StoW74 Stonebraker, Michael & Wong, Eugene Access control in a relational data base management system by query modification. ACM74 180-186 [x Retrieve access is controlled by automatically ANDing access restrictions to the user's stated retrieval conditions prior to their interpretation. x] <relational data base, access control; query modification>

Tsic72 Tsichritzis, Denis System Security. IBM Thomas J. Watson Research Center RC 3989 72.8.17

Tsic73 Tsichritzis, D. Reliability. Lecture Notes in Computer Science Vol. 30: Advanced Course on Software Engineering F.L. Bauer (ed.) Springer-Verlag 1973 319-373 [x Introductions to basic concepts of protection and security. x] <protection vs. security; domain; monitor; unique names; capabilities; revocation; indirect capabilities; mutually suspicious processes; access matrix; access control list; data dependent access>

Tsic74 Tsichritzis, D. A note on protection in data base systems. IWPOS74 243-248 [x Problem of <data base protection> is too complex to be solved by current operating system protection mechanisms. x] <capabilities, inadequacies; capability procedures; protection rings, inadequacies>

Ulha75 Ul Haq, Mohammed Inam Insuring individual's privacy from statistical data base users. NCC75 941-946

Vand69 Vanderbilt, Dean Hanawalt Controlled information sharing in a computer utility. (Thesis) MIT Project MAC TR-67 69.10.24

Wal+74a Walter, K.G.; Ogden, W.F.; Rounds, W.C.; Bradshaw, F.T.; Ames,

S.R. & Shumway, D.G. Primitive models for computer security. Case Western Reserve Univ. Department of Computing and Information Sciences ESD-TR-74-117 74.1.23 36p NTIS: AD-778 467 [x Model of <governmental security>, and modified version in which the <repositories> are directories and files of a tree-structured file system. x] <agents; need to know; security classes; classification; clearance; information transfer path; need to modify; discretionary/mandatory security>

Wal+74b Walter, K.G.; Ogden, W.F.; Rounds, W.C.; Bradshaw, F.T.; Ames, S.R.; Biba, K.J.; Gilligan, J.M.; Shaeffer, D.D.; Shaen, S.I. & Shumway, D.G. Modeling the security interface. Case Western Reserve Univ. Dept. of Computing and Information Sciences Report No. 1158 74.8 130p [x Approach to the design of a <security kernel> for <military security> in Multics, by a process of successive refinement of models, and corresponding proofs that security assumptions continue to hold for each successive model. x] <security system; security perimeter; repositories; agents; classification; clearance; information transfer path; discretionary/mandatory security>

Wal+75 Walter, K.G.; Schaen, S.I.; Ogden, W.F.; Rounds, W.C.; Shumway, D.G.; Schaeffer, D.D.; Biba, K.J.; Bradshaw, F.T.; Ames, S.R. & Gilligan, J.M. Structured specification of a security kernel. SIGPLAN Notices 10,6(75.6) (ICRS75) 285-293 [x Design methodology for a <governmental security> system: successive refinement of models, about which basic theorems can be proved. x] <security kernel; information security, models; information flow monitoring; security system; classification; clearance; sensitivity level; repositories; agents; transfer relation/path; executors; alter list; view list>

Walk73 Walker, R.D.H. The structure of a well-protected computer. (Thesis) Cambridge Univ. 1974

Weis69 Weissman, C. Security controls in the ADEPT-50 time-sharing system. FJCC69 119-133 [x File access is granted if and only if the (accumulated) <authority level> of both the user and his terminal is greater than that of the file, and the <category> of both user and terminal includes that of the file, and the user is a member of the <franchise> of the file. x] <authority history; high water mark>

Weis75 Weismann, Clark Secure computer operation with virtual machine partitioning. NCC75 929-934

Whit75 White, J.C.C. Design of a secure file management system. Mitre Corp. MTR-2931 ESD-TR-75-57 75.4 29p NTIS: AD-A010 590 [x Preliminary design of a file management system intended to operate under the Mitre PUP-11/45 <security kernel>. x] <access control list; x-property; access control, hierarchical directory structure; semaphores, access restrictions>

Wul+73 Wulf, W.; Cohen, E.; Corwin, W.; Jones, A.; Levin, R.; Pierson, C. & Pollack, F. Hydra: the kernel of a multiprocessor operating system. CACM 17,6(74.6) 337-345 [x Salient features are the dynamic type/object hierarchy and the protection mechanisms for procedure activations and

parameter passing. *) <kernel; object; local name space; capability; call mechanism; protection vs. security; ownership; privilege hierarchy; type; kernel rights; auxiliary rights; parameter template; subsystem; walk right>

Zill73 Zilles, Stephen N. Procedural encapsulation: a linguistic protection technique. SIGPLAN Notices 8,9(73.9) (PLOS73) 142-146
[* Objects and types are characterized only by (the procedures that implement) the operators defined on them; the protection features required for such <procedural encapsulation> are satisfied by <domain architectures>. *]

INDEX

*-property Bell73b Lipn75 Mill75 Neu+77 Scha75 Whit75
 absolute vs. defensive protection Lam+77
 abstract data types AmbH77 LisZ74
 access Jone73
 access attributes Bell73a Bell73b GraD72 Lamp71
 access attributes (types) Down73
 access bracket Grah68
 access categories Mart73
 access category IBMx71
 access clique BroS71
 access code Denn76b Neu+77
 access codes Elli74 Fabr68
 access condition HarH75
 access control BroS71
 access control features, language AmbH77
 access control limitations Mins76a
 access control list GraD72 Pope73a Pope74 SalS75 Salt73 Smit74
 Tsic73 Whit75
 access control lists Brat75 Con+72b DalN65 Down73 KarS74 Lamp69
 Rote74 Schr72
 access control lists vs. capabilities Salt76
 access control model HarH75
 access control packet Rote74
 access control procedure, per object Lamp71
 access control, container vs. data Salt76
 access control, data base Cha+75 Hel+75
 access control, hierarchical directory structure Whit75
 access controller SalS75
 access decision timing HarH75
 access functions Down73
 access graph Pope73a
 access history HarH75
 access key Lamp69 Lamp71
 access keys LamS75
 access list Denn76b
 access lists BroS71
 access lock list Lamp71
 access map Pope73a
 access matrices Bell73a
 access matrix Bell73b Con+72a Jone73 Lamp71 Pope74 Tsic73
 access matrix model GraD72 Har+75 Lind76b
 access matrix model, limitations HarH75
 access matrix, extended Fer+75
 access modes Lind76b
 access modes, primitive Salt73
 access monitor Down73
 access path control Eval67

access paths DaIN65 Pope73a
 access predicates Fer+75 PopK74a
 access relation Down73
 access right Lind76b
 access rules, requirements GraD72
 access types DaID74 Smit74
 access vector Rob+75
 access, static/dynamic Schr72
 accesses, normal vs. security update Down73
 accessible sets PopK74b
 accounting CohJ75
 accounting channel Rote74
 accumulations Ames74
 accuracy measure Jone73
 activation/incarnation mechanism Spi+74
 active object Pope74
 actor, process/procedure Andr74
 actors Pope73a
 address map, control information Eval67
 administer (access type) Fer+75
 agents Ames74 Fer+74 Wal+74a Wal+74b Wal+75
 aggregation Ande72b
 aliases CohJ75
 allocate (access type) Fer+75
 allocation/deallocation residuals HoIB76
 alter list Wal+75
 alter relation Ames74
 alteration principle Neu+77
 amplification CohJ75 HarH75 Jone73 JonL76b JonW75 Lind76b
 amplification, authorization Rede74
 amplifrights JonW75
 application rules Mins76a
 appropriateness, policy JonW75
 argument checking Salt73
 argument passing Rote74
 argument spying Rote74
 associative programs IBMx71
 asynchronous events Need73
 asynchronous i/o Bran73
 asynchronous i/o vulnerability Ande72b
 asynchronous interrupt vulnerability Ande72b
 asynchronous modification of arguments Grah68
 asynchronous modification of instructions BelW74
 attenuation of privilege Denn76b
 attributes, access/control Andr74
 authentication Lamp74
 authentication (type checking) Morr73b
 authentication, file login program Hsia68
 authority BroS71
 authority hierarchies Rote74
 authority hierarchies, undesirability Frie70
 authority hierarchy Salt73

authority history Weis69
 authority item Hoff71 Hsia68
 authority level Weis69
 authority list GraD72
 authority structures SalS75
 authorization BroS71 Brow71 IBMx71
 authorization classes DalD74
 authorization level Mart73
 authorization process HarH75
 authorization specification, validity/consistency HarH75
 authorization structures Mart73
 authorization tables, user/data Mart73
 authorization types, data base Cha+75
 authorization, def. Frie70
 authorization, files Ande72b
 authorized program facility McPh74
 authorized subspace HarH75
 auxiliary invocation, domain HarH75
 auxiliary rights Wul+73
 base level (kernel) Rede74
 basic monitor Andr74
 benign domain Rote74
 billing Rote74
 binding rules JonL76b
 binding, static/dynamic Denn76a
 blind service Rote74
 brands Mins76a
 breaches, internal vs. external Down73
 c-list CohJ75 DenV66 GraD72 Neu+77 Pope74
 c-lists LamS75
 call and return mechanism Neu+77
 call bracket Grah68
 call mechanism Wul+73
 call packet Rote74
 call-in Rote74
 call-out Rote74
 call/return operations, domain Fer+74
 calls, cross-ring Grah68
 capabilities Andr74 CohJ75 Fabr68 Lamp69 Lamp74 LamS75 Lind73
 Lind76b Need73 Neu+77 Pope74 Rob+75 Rote74 SalS75 Tsic73
 capabilities (variables) JonL76b
 capabilities and storage allocation Fabr74
 capabilities vs. pointers Coss74
 capabilities, control of transfer SevT74
 capabilities, extended JonW75
 capabilities, inadequacies Tsic74
 capabilities, indirect, chains Rede74
 capabilities, passing of Rote74
 capabilities, potential Moor74
 capabilities, protection of Rede74
 capabilities, representation integrity, approaches Fabr74
 capabilities, sealed Rede74

capabilities, user Fabr68
 capability Jans74 Jone73 Schr72 Wul+73
 capability (training) Mart73
 capability architecture Coss72 Engl72 Engl74
 capability architectures Denn76b
 capability channels Neu+77
 capability faulting PopK74a
 capability list Denn76b DenV66 Lamp71
 capability lists Rote74
 capability manager Rob+75
 capability procedures Tsic74
 capability registers Engl72 Fabr68
 capability registers, real/virtual Engl74
 capability segment NeeW74
 capability segments Coss74 Fabr68 Need72
 capability systems, goals Rede74
 capability transfer operations JonW75
 capability vector mechanism Saal77
 capability-based addressing Fabr74 Lind76b
 Capability-based protection mechanism Andr74
 capability-based system, extensible Fer+74
 caretaker domain Rede74
 caretaker procedure Spi+74
 caretaker program Rote74
 case/uncase operators Fer+74
 categories BroS71
 categories (access privileges) BeIL73b
 categories, user/data Mart73
 category Scha75 Weis69
 central capability segment Engl74
 certification mechanism DenD77
 changes relation Ames74
 channel programs Salt73
 channel programs, vulnerabilities BelW74
 channels, legitimate/covert Mill75
 chattels Lamp74
 checking, dynamic vs. static Jone73
 checkrights JonW75
 classes AmbH77
 classes, Simula 67 Palm73
 classification BroS71 Scha75 Wal+74a Wal+74b Wal+75
 classification basis, record/field Scha75
 classification systems Lind76b Salt76
 classification, derived Ande72a
 classifications BeIL73a BeIL73b
 clearance Ames74 Scha75 Wal+74a Wal+74b Wal+75
 clique, user Frie70
 closure, confidentiality Den+74
 cluster of operations LisZ74
 clusters Ames74
 collusion Pope73a
 column system Con+72a

communication by value vs. pointer Lamp74
 communication, interprocess/interdomain Rede74
 compartment Scha75
 compartmentalization Frie70
 compartmentalization (isolation) Mart73
 compartments Down73 Salt73
 compile time access control Con+72a
 compile-time checking Con+72a Con+72b Den+74 Fer+75 JonL76b
 compile-time enforcement MooC74
 compile-time/run-time enforcement Denn76a
 compiler trap doors KarS74
 completeness JonL75
 completeness requirement GraD72
 complexity of an access control mechanism Spie74
 comprehensibility of protection mechanisms Rede74
 confidence modules Mins76a
 configuration, protection system Har+75
 confined computation, def. Andr74
 confinement CohJ75 JonL75 Mins76a SaIS75 Salt76 SevT74
 confinement mechanism Den+74
 confinement of errors Denn76b
 confinement policy JonW75
 confinement principle Neu+77
 confinement problem Den+74 Lamp73 Lipn75
 confinement property DenD77
 conservation CohJ75
 consistency, policy Down73
 consistency, single variables Bis+75
 constituent rights Jone73
 constraints PopK74b
 constraints, updatation (policy) Down73
 constructive design Pope74
 contextual sensitivity BroS71
 control access Bell73b
 control attribute Down73 GraD72
 control profile, user Hoff71
 control protection Lamp67 Lamp68
 control transfers Lamp69
 control, transfers of Coss74
 copy flag GraD72 Lamp71
 copy flag, transfer-only GraD72
 copy of capability, literal/revocable RedF74
 courier policy JonW75
 covert channels Bran73 Den+74 Lamp73 Lipn75
 criticality Carl76
 cross-domain calls Schr72
 data access monitor Scha75
 data analysis, prevention Mins76a
 data base access control Fer+75
 data base privacy mechanisms GriW76
 data base protection JonW75 Tsic74
 data base scheme Car+71 Con+72b

data dependent access Tsic73
 data dependent conditions Con+72a
 data dependent decisions Hoff71
 data management systems Ande72b
 data mark machine Denn76a
 data security JonL75
 data security, def. Mart73 PopK74a
 data-dependent access decisions Pope74
 database protection Mins76a
 database subschema Mins76a
 debugging problem GraD72
 decision rule Con+72a
 decision sequences Bell73a Bell73b
 decision verification Denn76b
 decisions, prior/future CohJ75
 decisions, unilateral/negotiated CohJ75
 declared rights Jone73
 declassification Den+74
 education problem Palm74
 defensive programming Lind76b
 definition of controlled resources IBMx71
 degree of protection Elli74
 delete access Scha75
 denial of service Neu+77
 deny/allow descriptions Mano71
 discretionary access control matrix Mill75
 descriptor SalS75
 descriptor (capability) Fer+74
 descriptor-based architecture Ande72b
 descriptors Ande72a Brat75
 descriptors, privacy Denn76b
 descriptors, protection Brat75
 design principles SalS75
 detectability Brat75
 detection principle Neu+77
 diagonal system Con+72a
 direct access predicate PopK74a
 directories Lind76b Salt73
 directories in capability system Rede74
 directories, protection Brat75
 directory initiation Brat75
 directory of authorized users Con+72b
 discretionary controls SalS75
 discretionary/mandatory security Wal+74a Wal+74b
 discretionary/nondiscretionary controls Lind76b
 disengagement Den+74
 distinguished entry Neu+77
 domain Lamp71 Lamp74 Pope74 SalS75 Schr72 Tsic73
 domain (protected subsystem) Spi+74
 domain architectures Zill73
 domain changing Denn76b
 domain of authorization HarH75

domain switching Lind76b
 domains CohJ75 Fer+74 GraD72 Lamp69 LamS75 Lind76b Neu+77 SchS72
 domains and processes Rede74
 domains, postulates Rote74
 dominance, access history HarH75
 domination Scha75
 dynamic authorization SalS75
 dynamic linker Jans74
 dynamic rights Jone73
 dynamic type creation Jone73
 elimination rules Mill75
 encapsulated type Fer+74
 encapsulation LisZ74 Rote74 SalS75
 encapsulation, process Denn76b
 enforcement process HarH75
 enforcement rule Jone73 Pope74
 engagement Den+74
 enter access Engl72 Fabr68
 enter access/instruction Fabr74
 enter capability Denn76b Engl74 Need73 Neel74
 enter right Lind76b
 entry, protected Eval67
 environment Andr74
 environment binding Pope74
 environment binding rule Jone73
 environment crossing operations JonW75
 environment crossing operators Jone73
 environment of domain Fer+74
 environment representations Jone73
 environment, open/closed Denn76b
 environments Jone73
 error conditions Den+74
 error decision Bell73b
 error patterns Palm74
 error types Lack74
 errors, examples Bis+75 Carl76 HolB76 Pope74
 errors, integrity, taxonomy/categories/examples Abb+76
 errors, types/examples Mcph74
 excessive use of resources SevI74
 execute-only access DalD74
 executors Ames74 Wal+75
 existence, knowledge of PopK74a
 extended resources HarH75
 extended-type manager Lind76b
 extended-type objects Lind76b
 extender capability Rede74
 failsafe principle SalS75
 field level control Car+71 Con+72b Mano71
 field, protected Frie70
 file access control Gain72 Hsia68
 file access control scheme Bar+67
 file access via given program Need73

file authorization Ande72b
 file backup Salt73
 file control code Bing65
 file directories DaIN65
 file groups DaID74
 file keys Palm74
 file management kernel PopK74a
 filtering policy JonW75
 filtering property Pope73a
 flow condition Moor74
 flow relation/policies DenD77
 formal specification Burk74
 formularies Pope74
 formulary Hoff71
 franchise Weis69
 franchise, user/operation/resource unit HarH75
 freezing CohJ75
 functional access Fer+75
 gate Jans74
 gatekeeper Grah68 JonW75 KarS74
 gates Lamp69 Salt73 Schr72 SchS72
 generic weaknesses Bran73
 global object name systems Laue74
 goals, protection mechanisms Lind74
 governmental security Wal+74a Wal+75
 grain of protection Pope74
 grant Cha+75
 grant command GraD72
 grants, labeled GriW76
 group (protection category) Frie70
 grouping of users and resources IBMx71
 hidden channels Neum73 Pope74
 hidden data Rote74
 HIDDEN specification Palm73
 hiding, information Lipn75
 hierarchical access control Ande72b DaIN65
 hierarchical control SalS75
 hierarchical process structure NeelW74
 hierarchically structured operating system DonM75
 hierarchically structured systems ChaS76
 hierarchies, subject, advantages GraD72
 high water mark BroS71 JonL75 Lipn75 Scha75 Weis69
 higher level errors Pope74
 i/o channels Ande72a
 i/o operations Salt73
 identification Brow71 Lamp71
 identification, objects McPh74
 identification, subjects GraD72
 identifiers, unique Rede74
 implied sharing Ande72a
 implied sharing vulnerability Ande72b
 incomplete parameter checking Ande72b

indirect access GraD72 MooC74
 indirect access predicate PopK74a
 indirect capabilities SevT74 Tsic73
 indirect information paths Mill75
 indirection Lind76b
 indirection and control RedF74
 indirection table NeelW74
 indirection tables Need72
 inference Ande72b Neu+77 Pope74 Salt76
 inference problem Pope73b
 influencability/influentiality Carl76
 information control vs. access control JonL75
 information flow DenD77
 information flow graph MooC74 Moor74
 information flow monitoring Wal+75
 information flow requirements Saal77
 information flow, control Salt76
 information flow, explicit/implicit Denn76a
 information security, models Wal+75
 information structures, control/computing Andr74
 information transfer path Wal+74a Wal+74b
 inherit (access type) Fer+75
 initialization CohJ75
 input validation Brow71
 input/output PopK74a
 integrity constraint Hel+75
 integrity, data Brow71
 integrity, data base Hel+75
 integrity, data/system Smit74
 integrity, def. ChaS76 DonM75
 integrity, system Frie70 Mart73
 intent specification Salt73
 intention, explicit Atta73
 intentional resolution Mins76a
 inter-module protection Palm73
 interactivity principle BelL73b
 intermediaries Lind76b
 intermediate results, hiding Mins76a
 interruptibility PopK74a
 isolation BelW74 Brow71 Fabr73 Lamp74 Spie74
 isolation level Elli74
 isolation mechanisms Ande72a
 isolation, authorization mechanism Frie70
 isolation, def. ChaS76
 isolation/interaction SevT74
 item code Car+71
 item key Car+71
 kernel Fer+74 Pope73b Spi+74 Wul+73
 kernel rights Wul+73
 kernels, levels PopK74a
 key GraD72
 key variables Pope73a

lattice model Den077
 layers of protection, advantages Grah68
 leakage, information Neu+77
 leaking, rights Har+75
 least common mechanism Schr75
 least privilege SalS75
 least privilege principle Saal77 Salt73
 legitimate channels Lipn75
 levels of classification, undesirability Frie70
 levels of mechanism PopK74a
 levels of protection GraD72
 levels of representation Burk74
 limited-use systems Salt76
 link fault handling Jans74
 list-oriented mechanisms SalS75
 load/store, capabilities EngI74
 local name space JonW75 Wul+73
 lock functions Pope73a
 lock list GraD72
 lock-and-key mechanism Con+72b
 lock-and-key scheme Gain72
 locked boxes Pope74
 locker capability Rede74
 locks and keys Down73 Pope73a Pope74
 locks, data record Mart73
 locksmith Rote74 Salt73
 lockwords Mart73
 login program, file Hsia68
 looks-at relation Ames74
 lost object problem Neu+77 Rede74
 lost objects CohJ75
 lying to prevent detectability Brat75
 manager problem Andr74
 mandatory/discretionary controls Ames74
 manipulators Ames74
 masking, channels Lamp73
 master mode KarS74
 matrix, user-privilege Frie70
 maximally complete mechanism JonL75
 mechanism, def. Andr74 JonL75
 mechanisms, def. Need73
 mechanism requirements ChaS76
 mechanisms, functions JonW75
 mediated access Neu+77
 memory environment Lamp74
 memory protection Lamp67 Lamp74
 memoryless operation Neu+77
 memoryless procedure Jone73
 memoryless process Pope74
 memoryless specifications HarH75
 memoryless subsystems GraD72 JonL75
 memorylessness Lamp73

memorylessness and complexity Spie74
 message confinement Andr74
 message model Lamp74
 message system Fabr73 Lamp71 Rote74
 military security Wal+74b
 military security classification Neu+77
 military security model Down73
 modification CohJ75
 modify relation Ames74
 modularity Lind76b PopK74b
 modules AmbH77
 monitor Tsic73
 monitor, object type GraD72
 monitoring policy JonW75
 monitors Andr74
 Morse Code problem PopK74a PopK74b
 move-without-reading Neum73
 multi-level security KarS74
 multiple domain processes Denn76b
 mutual exclusion, customer processes Den+74
 mutual suspicion Andr74 CohJ75 Fer+74 Lind76b
 mutually suspicious processes Tsic73
 mutually suspicious subjects Neu+77
 mutually suspicious subsystems GraD72 Pope74
 name space model Jans74
 name spaces Rote74
 name, local/system Denn76b
 names, concealment Frie78
 names, scope/sharing/protection GeoS73
 naming hierarchy Rote74
 need to know Down73 IBMx71 Lipn72 Mart73 Neu+77 Scha75 Wal+74a
 need to modify Wal+74a
 need-to-know categories Bell73a
 need-to-know principle Grah68 Jone73 ParP74
 nested address space systems Laue74
 nonleakage Den+74
 nonretention Den+74
 numerical measure of security Lam+77
 o operators JonL75
 o-functions Lipn75 Mill75
 object Jans74 Jone73 Wul+73
 object system Lamp71
 objecthood PopK74a
 objects BroS71 Down73 Fer+74 GraD72 JonW75 Lind76b Neu+77
 observability postulate JonL75
 observe relation Ames74
 operator-type restrictions SalS75
 operators, non-monadic Rede74
 operators, strong/weak Fer+74
 own right Har+75
 owner Hsia68 PopK74b
 owner attribute Down73 GraD72

ownership CohJ75 Jone73 Lamp71 Scha75 Wul+73
 ownership of domains Rote74
 ownership privilege Rede74
 ownership, absolute/conditional HarH75
 parameter checking Pope74
 parameter checking, incomplete Bran73
 parameter passing AmbH77 Bis+75 Jone73
 parameter spaces Eval67
 parameter template Wul+73
 parameters, revocable Rede74
 part owner Bar+67
 partial rejection HarH75
 passive object Pope74
 passwords, file Frie70 Lamp69 Mart73
 path condition Moor74
 penetration Pope74
 penetration techniques KarS74 Lack74
 permission principle Salt73
 policy, def. JonL75 PopK74b
 policy vs. mechanism Jone73
 policy, informality Salt76
 prescript SalS75
 primitives, protection state/environment Andr74
 principal SalS75
 principals Salt73
 principle of control Andr74
 principle of least privilege Jans74 Lind76b Pope74
 principle of maximum security Neum73
 privacy Brow71
 privacy (security) Mins76a
 privacy restrictions Bell73b Rote74
 privacy, def. ChaS76
 privilege hierarchy Wul+73
 privilege levels Smit74
 privilege list BroS71
 privilege number Denn76b
 privilege state mechanism Denn76b
 privilege structure, complexity Frie70
 privileged code, exclusion from user process PopK74a
 privileged instructions, deficiency Grah68
 privileged mode, nonnecessity Engl72 Engl74
 privileges, program/user Frie70
 procedural controls Ande72b
 procedural embedding CohJ75
 procedural encapsulation Zill73
 procedural vulnerabilities KarS74
 process control as capability privilege Rede74
 process isolation Denn76b
 program authorization McPh74
 program identity Ande72a
 program reference table Bing65
 program restricted data Mart73

program-restricted data sets IBMx71
 progressive authorization HarH75
 propagation CohJ75 SalS75
 propagation of protection specifications Salt73
 propagation, knowledge Rede74
 property lattice DenD77
 proprietary programs Lamp69
 proprietary services Rote74
 proprietary services, maintenance Rote74
 protected entry Denn76b
 protected procedure Lind76b NeeW74
 protected subroutines Coss72
 protected subsystem Jans74 Salt73 Schr72
 protected subsystems DalD74 Den+74 SalS75 Salt76
 protection, def. Jone73 Lamp74
 protection as restriction Neum73
 protection categories, file/process, read/write Gain72
 protection data Pope74
 protection group SalS75
 protection groups BroS71
 protection principles, languages AmbH77
 protection problems CohJ75
 protection specification language HarH75
 protection state Andr74 GraD72
 protection state, transformations, closure Jone73
 protection systems, comparison Jone73
 protection theorems Rob+75
 protection vs. security Tsic73 Wul+73
 protection, absolute/defensive Lamp74
 protection, abstract formulation Rote74
 protection, access/control CohJ75
 protection, def./objectives Schr72
 protection, logical vs. physical Andr74
 psychological acceptability SalS75
 qualified type JonL76b
 query modification Hel+75 Stow74
 question decision BelL73b
 read-without-copying Neum73
 record code Car+71
 record key Car+71
 record level control Car+71 Con+72b Hsia68 Mano71
 redundant checking Pope74
 redundant security DonM75
 reference monitor Ande72b Burk74 Lipn72
 reference name space Brat75
 regime of protection Need73
 registrar Lamp74
 relational data base, access control Stow74
 repositories Ames74 Wal+74a Wal+74b Wal+75
 request sequences BelL73a BelL73b
 requirements, protection mechanisms JonW75
 residual, content/access HoIB76

residue Bran73
 resource control Denn76b
 resources as protected data structures Eng174
 restrict statement Hel+75
 restricted names McPh74
 restriction graph Pope73a
 review of access SalS75
 revocability GraD72
 revocability of revocability RedF74
 revocability, revocable Rede74
 revocable copy Neu+77
 revocation Cha+75 Lamp71 Lind76b Pope74 Rede74 Rote74 SalS75 SevI74
 Tsic73
 revocation of capabilities RedF74
 revocation policy JonW75
 revocation, immediate/permanent/selective/partial/temporal/sharing CohJ75
 revocation, recursive GriW76
 revocation, selective Neu+77
 revoker capability Rede74
 right transfer rule/primitives Jone73
 right (part of application rule) Mins76a
 rights transfer primitives Pope74
 rights, generic Har+75
 rights, generic/auxiliary CohJ75
 rights, kernel/auxiliary JonW75
 rights, quantitative SevI74
 rights, required/new CohJ75
 ring bracket Grah68
 ring brackets, read/write/execute SchS72
 ring crossing/structure SchS72
 rings Lamp69 Neu+77 Pope74
 rings, protection Brat75 Grah68 Jans74 KarS74 Salt73 Schr75
 rings, protection, inadequacies Tsic74
 rule, def. BeIL73b
 rules of operation BeIL73b
 safety Har+75
 scavenging problem Ande72b
 scheduling PopK74a
 scope of domain Lamp74
 scope of names Laue74 Pope74
 scope rules AmbH77
 seal operators, opaque/transparent Morr73b
 seal/unseal CohJ75 Pope74
 sealed capabilities Rede74
 seals LamS75 Morr73a
 secrecy (of representations) Morr73b
 security by association Mart73
 security class, field Con+72b
 security classes Ames74 DenD77 Denn76a Wal+74a
 security clearances Denn76a
 security condition BeIL73b
 security enhancements, types Abb+76

security events/axioms Ames74
 security kernel Ande72b Burk74 ChaS76 Down73 Mill75 Neu+77 Pope74
 PopK74b Rob+75 Schr75 Wal+74b Wal+75 Whit75
 security kernel, criteria Jans74
 security kernel, functions PopK74a
 security kernel, name management functions Brat75
 security kernels PopK74a
 security levels IBMx71 Mart73
 security matrix MooC74
 security object Pope74
 security objects Pope73a PopK74a PopK74b
 security of information, def. Need73
 security officer DalD74 IBMx71 Mart73
 security perimeter Wal+74b
 security policy, def. JonW75
 security primitives PopK74b
 security principle Bell73b
 security problems, special Rob+75
 security sensitive instructions PopK74a
 security space HarH75
 security state Down73
 security system Pope73b Wal+74b Wal+75
 security vs. privacy Con+72a
 security, def. Bell73a BellW74 ChaS76 DonM75 Lind76b Rob+75
 segment SalS75
 segment as unit of protection Brat75 Coss74 Denn65 DenV66 Fabr68
 Grah68 Need73 Schr72 SchS72
 segment descriptor Salt73
 segment, basis for access control Lipn72
 segments Rote74
 selective access AmbH77
 selective confinement Den+74
 self control SalS75
 semaphores, access restrictions Whit75
 sensitivity level Wal+75
 sensitivity levels SalS75
 separation of privilege SalS75
 serialization mechanisms McPh74
 service program Lamp73
 set-theoretic model Pope73b
 sharing, information vs. other resources BellW74
 sharing, arbitrary patterns Salt76
 sharing, slow/fast Lamp74
 simple security condition Mill75
 single-tag rule Frie70
 soundness JonL75 JonL76a Rote76
 sphere of protection Pope74
 spheres of protection Denn64 Denn65 DenV66
 sponsor Sev74
 state sequences Bell73a Bell73b
 static privacy conditions Con+72a
 static storage allocation Jans74

statistical access Pope74
statistical access problem Pope73b
storage capability Denn76b
storage channels Lipn75
storage protection McPh74
storage residue Salt73
stratification Mart73
subfile access control Smit74
subject Jans74
subjects Down73 GraD72 Lind76b Neu+77
subletting Rede74
subownership HarH75
subsystem Jone73 Wul+73
subsystems CohJ75
subverter program KarS74
suitability factor Jone73
supervisor as interaction controller Fabr68
surveillance mechanisms JonL75
suspicion SevI74
synchronization Pope74
system administrator HarH75 Salt73
system initialization Jans74 Salt73
system integrity McPh74
tagging Frie70
template JonW75
templates CohJ75
ticket-oriented mechanisms SalS75
time problems Mill75
time-of-check-to-time-of-use error Bis+75
time-of-check-to-time-of-use problem McPh74
trademarks Lamp74 Morr73a
tranquility principle Bell73b Mill75
transaction types Mart73
transfer command GraD72
transfer relation/path Wal+75
transition type Fer+74
transitivity of confinement Lamp73
transitivity of engagement Den+74
trap doors, classes KarS74
trap extension Salt73
trap list DaIN65
trojan horse attack Bran73
trojan horse attacks Neu+77
trojan horse problem Ande72b Lind76b Need73
trust GraD72
trust, changing levels Rede74
trustworthy subjects GraD72
two state operations Ande72a
two-mode systems, deficiency Grah68
type Jone73 Wul+73
type checking LisZ74
type extendibility Rede74

type manager Neu+77 Rob+75
 type transformation Fer+74
 type-module JonL76b
 undecidability, safety Har+75
 unique names Tsic73
 universal subject GraD72
 untrustworthy subsystems GraD72
 update monitor Down73
 update program Pope74
 update rules, access matrix GraD72
 updater PopK74a PopK74b
 usage attributes DalN65
 use (access type) Fer+75
 user categories Smit74
 user control profile Bing65
 user data passed as system data McPh74
 user descriptor Grah68
 user groups DalD74
 user interface Neum73
 user profile Frie70
 user responsibility Down73
 user-supplied addresses McPh74
 utility programs, dump/patch KarS74
 v functions JonL75
 v-functions Lipn75 Mill75
 validation errors Carl76
 validation of arguments, insufficient KarS74
 validation of inputs HarH75
 validity checking McPh74
 validity vs. integrity Carl76
 value dependent authorization Cha+75
 view list Wal+75
 view relation Ames74
 views GriW76
 views, data base Cha+75
 virtual I/O Smit74
 virtual machine monitor DonM75
 virtual machines Ande72a Denn76b Fabr68 Pope74 PopK74a PopK74b
 Smit74
 virtual machines, security advantages Atta73
 virtual memory Smit74
 virtual memory as protection mechanism ParP74
 virtual time Lipn75
 virtual user Con+72a
 volume, foreign/local SevT74
 vulnerabilities, classes Ande72b
 vulnerabilities, descriptions KarS74
 vulnerabilities, multilevel PopK74a
 walk right Wul+73
 withdrawal of rights HarH75
 zones, data Mart73